An ExpTime Procedure for Description Logic $\mathcal{ALCQI}$ (Draft)

Yu Ding

Concordia University, Montreal QC H3G 1M8, Canada
ding_yu@cse.concordia.ca

Document Created: Aug-10-2006
Latest Modification: Jan-11-2007

Abstract. A worst-case ExpTime tableau-based decision procedure is outlined for the satisfiability problem in $\mathcal{ALCQI}$ w.r.t. general axioms.

1 Motivation and Brief Introduction

The *concept satisfiability problem* in description logics (DLs) with both $\mathcal{Q}$ and $\mathcal{I}$ has been considered empirically the hardest of all for those DL problems in the ExpTime complexity class. Though the C-rule (the Ramsey's Rule)[Din07] works for other logics like $\mathcal{ALCFI}$ or $\mathcal{ALCOI}$, it is not obviously applicable to DLs with the *qualified number restrictions*. In this paper, we take a different and general approach for $\mathcal{ALCQI}$. The focus is an ExpTime tableau-based procedure and therefore empirical issues are not concerned. We start with a brief introduction to the DL $\mathcal{ALCQI}$, the *general inclusion axioms*, and the *concept satisfiability problem*. For more we refer to [BCM+03].

Definition 1. (Concept Formulae) *We use A for* atomic concept*, use C and D for arbitrary concepts, use R for a role name. For non-negative integer n, concept formulae in $\mathcal{ALCQI}$ are formed according the following grammar*[1]*:*
$C, D := \top | A | \neg C | C \sqcap D | C \sqcup D | \exists^{\leq n} R.C | \exists^{\geq n} R.C$

Definition 2. (Semantics) *An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ consists of a set $\Delta^{\mathcal{I}}$ (the domain) and an interpretation function $.^{\mathcal{I}}$. The interpretation function maps each concept name C to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role name R to a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Let the symbols C, D be concept formulae, R be a role name. The interpretation function can be inductively defined as follows:*

$\top^{\mathcal{I}} := \Delta^{\mathcal{I}}$ $\quad$ $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$ $\quad$ $(C \sqcup D)^{\mathcal{I}} := C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$(\exists^{\leq n} R.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} \mid ||\{y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}|| \leq n\}$
$(\exists^{\geq n} R.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} \mid ||\{y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}|| \geq n\}$
and additionally, it satisfies $(x,y) \in R^{\mathcal{I}} \Leftrightarrow (y,x) \in (R^{-})^{\mathcal{I}}$.

[1] W.l.o.g. $\exists R.C$ is expressed in $\exists^{\geq 1} R.C$, and $\forall R.\neg C$ is expressed in $\exists^{\leq 0} R.C$.

Definition 3. (Negation Norm Form) *The negation normal form is defined by applying the following transformation in such a way that negation signs are pushed inward and appear only in front of concept names.*

$\neg\neg C \rightarrow C$ $\qquad$ $\neg(C \sqcap D) \rightarrow \neg C \sqcup \neg D$

$\neg(C \sqcup D) \rightarrow \neg C \sqcap \neg D$ $\qquad$ $\neg\exists^{\leq n} R.C \rightarrow \exists^{\geq n+1} R.C$

$\neg\exists^{\geq n} R.C \rightarrow \exists^{\leq n-1} R.C$

Definition 4. (Generalized Concept Inclusions) *If C is a concept formula, then $\top \sqsubseteq C$ (generalized concept inclusion or GCI) is a terminological axioms. A finite set of terminological axioms $\mathcal{T}$ is called a Tbox. The interpretation function $\cdot^{\mathcal{I}}$ is extended for GCI as $(\top \sqsubseteq C)^{\mathcal{I}} := \top^{\mathcal{I}} \subseteq C^{\mathcal{I}}$. Without lose of generality, the general inclusion axioms can be expressed in one bigger GCI in NNF.*

2 Preliminaries and Notations

In the paper[2] we call $\exists^{\leq n} R.C$ and $\exists^{\geq n} R.C$ *modal constraints*, and call $C \sqcap D$ and $C \sqcup D$ *propositional constraints*. We assume each role has a unique inverse role. For a role R, for example, we consider R^- as the only *inverse role*[3].

The discussion is put in the context of labeled trees. Each node is labeled with a set of concept formulae, each edge is labeled with a role[4]. What is important is to each (tableau-tree) node x we also attach algebraic objects like *systems of linear integer inequalities* (LIIs) $lii(x, R)$, and to each R-edge (to x's successors) we attach one *non-negative integer solution* $S(x, R)$ of $lii(x, R)$.

We basically require that readers are familiar with *propositional logic* and *integer linear programming*[Vas83][Sch86] (plus a bit knowledge of integer *matrix* and *linear algebra*). Several notions are to be explained below.

2.1 Cut Formulae

Definition 5. (Cut Formulae) *Give a concept E and a GCI G in $\mathcal{ALCQI}$ for satisfiability test. For each modal subformula of G and E of the form $\exists^{\bowtie n} R.C$, where R is any role, there is one cut formula as:*

($\star$) $\exists^{\leq 0} R^-.\top \sqcup C \sqcup \tilde{C}$.

The set of all cut formulae for E and G is denoted as $\mathcal{K}_a$.

The set $\mathcal{K}_a$ is trivially satisfiable[5] in any model for E and G. The most important to notice is that, due to the cut-formulae, the calculus can treat R^- and R as independent role names as if they had no inverse relationship at all. When this is exploited in the tree-like tableaux structure, the construction can be performed top-down and each node will be visited only once.

[2] For brevity, $\exists^{\bowtie n} R.C$ denotes $\exists^{\leq n} R.C$ or $\exists^{\geq n} R.C$, and $\tilde{C}$ is the NNF of $\neg C$.

[3] It takes a linear cost to identify equivalent role names that are implied by the declarations of inverse relationship in a namespace (of role names).

[4] The inverse relationship can be ignored due to the cut formulae introduced below.

[5] To be precise, any model for E and G can be extended to satisfy $\mathcal{K}_a$.

We denote as $\mathcal{C}(x)$ the result from splitting the cut formulae at x's R^--predecessor node and simply call it the *cut-set* for x. For a cut-formula $\exists^{\leq 0} R^-.\top \sqcup D \sqcup \tilde{D}$ at x's R^--predecessor, either $D \in \mathcal{C}(x)$ or $\tilde{D} \in \mathcal{C}(x)$.

2.2 Propositional Branch and Its Fine Tune

Definition 6. (Propositional Branches) *Give $\mathcal{L}(x)$ the set of labels for the node/element x, the propositional branches (PBs) for x is $\mathcal{BS}(x)$ the set of all possible disjuncts from the disjunctive normal form*[6] *(DNF) of $\mathcal{L}(x)$ by treating modal constraints as propositions. Denote the finite set of PBs as $\mathcal{BS}(x) = \{\mathcal{B}_1(x), ..., \mathcal{B}_i(x)...\}$.*

The notion of *propositional branches* (PBs) is quite intuitive if one considers the AND-OR structure of concept formulae and the results from exhaustively performing the $\sqcap$-rule and $\sqcup$-rule commonly seen in tableaux calculi such as for $\mathcal{ALC}$. Enumerating PBs for a set of labels means handling all outer $\sqcap$ and $\sqcup$ operators in this AND-OR structure (other than those located inside role fillers).

Definition 7. (Fine-Tuned Modal Constraints) *In the tableaux (labeled tree) $\mathtt{T}$, let x be the R^--predecessor of y, we have:*

- *Give $x \in C^{\mathcal{I}}$, then $y \in (\exists^{\leq n} R.C)^{\mathcal{I}}$ iff $\|\{z \in \Delta^{\mathcal{I}} : (y, z) \in R^{\mathcal{I}}$ and $z \in C^{\mathcal{I}}$ and z is R-successor of $y\}\| \leq n - 1$;*
- *Give $x \in C^{\mathcal{I}}$, then $y \in (\exists^{\geq n} R.C)^{\mathcal{I}}$ iff $\|\{z \in \Delta^{\mathcal{I}} : (y, z) \in R^{\mathcal{I}}$ and $z \in C^{\mathcal{I}}$ and z is R-successor of $y\}\| \geq n - 1$;*

These adjustments of cardinalities over successors depending on the cut-set chosen at the predecessor are called fine-tuning of modal constraints. We denote the propositional branch $\mathcal{B}(x)$ after fine-tuning as $\mathcal{B}'(x)$.

2.3 Linear Diophantine Inequalities

The procedure will be presented as in the *algebraic approach*. We reuse the *atomic decomposition* technique. What is typical of the algebraic approach[7] is the building of systems of LIIs from decompositions of role fillers on each role. For more we refer to Ohlbach's[OK99], Haarslev and Möller's[HTM01] work.

Definition 8. (Linear Integer Inequalities) *Linear (subset sum) integer inequalities (LII) is a system of special linear Diophantine inequalities (LDI) such that, for the finite set of variables $V = \{v_1, v_2, \ ..., \ v_j, \ ..., v_{2^\lambda - 1}\}$ from the non-negative integer domain, the k-th LDI is of the form $(\sum_{j=1}^{2^\lambda - 1} v_j \cdot w_{k,j}) \leq n_k$ or of the form $(\sum_{j=1}^{2^\lambda - 1} v_j \cdot w_{k,j}) \geq n_k$, where each constant $w_{k,j} \in \{0, 1\}$, each*

[6] We do not need a canonical (propositional) form and therefore DNF suffices. We treat each propositional branch as a set of modal constraints or concept literals.

[7] Regardless of the differences, the *atomic decomposition* and the special *linear integer inequalities* have intricate connections to the *choose-rule* and Tobies's *counter*.

unknown variable $v_j \in V$ is in the non-negative integer domain, and each n_k is some non-negative integer constant, λ is a non-negative integer constant. The number of unknown variables is $2^\lambda - 1$ where λ is the number of LDIs and is also the number of modal constraints before atom-decomposition.

3 The Decision Procedure for $\mathcal{ALCQI}$

For (tableau-tree) node x, we use $\mathcal{L}(x)$ for its initial label, and $\mathcal{B}(x)$ for its current *propositional branch*, and $\mathcal{B}'(x)$ for the corresponding *fine-tuned* one. The converted problem is E and $G \cup \mathcal{K}_a$ (in which the very special *cut-formulae* are contained). Below is a set of expansion rules for the converted problem.

PB-rule:	if	1. x is not blocked, and x is an R-successor, and
		2. $\langle \mathcal{C}(x), R, \mathcal{L}(x) \rangle \notin$ `Nogood`, and
		3. there is a $\mathcal{B}(x) \in \mathcal{BS}(x)$ such that
		(a) $\{\langle \emptyset, \epsilon, \mathcal{B}(x)\rangle, \langle \emptyset, \epsilon, \mathcal{B}'(x)\rangle\} \cap$ `Nogood` $= \emptyset$, and
		(b) $\langle \mathcal{C}(x), R, \mathcal{B}(x) \rangle \notin$ `Nogood`
	then	choose $\mathcal{B}(x)$ as the current propositional branch of x
LII-rule:	if	1. x is not blocked, and
		2. there are (modal constraints on R) $\exists^{\bowtie n} R.C \in \mathcal{B}'(x)$, and
		3. x has no LII for those modal constraints on R
	then	generate an LII for those modal constraints on R in $\mathcal{B}'(x)$, and generate upto $2^\lambda - 1$ atom-decompositions as R-successors

Fig-1. The tableaux expansion rules[8] for $\mathcal{ALCQI}$

The atom-decomposition for a set of *modal constraints* on a certain role generates all possible combinations about *role fillers* or *negated role fillers*. Each combination is considered as conjuncted together. Also see footnote 7. For example, for the set $\{\exists^{\leq 3} R.C_1, \exists^{\geq 2} R.C_2, \exists^{\geq 4} R.C_3\}$ of *modal constraints* on role R, the *atomic decomposition* is $\{C_1 \sqcap C_2 \sqcap C_3; C_1 \sqcap C_2 \sqcap \neg C_3; C_1 \sqcap \neg C_2 \sqcap C_3; C_1 \sqcap \neg C_2 \sqcap \neg C_3; \neg C_1 \sqcap C_2 \sqcap C_3; \neg C_1 \sqcap C_2 \sqcap \neg C_3; \neg C_1 \sqcap \neg C_2 \sqcap C_3\}$ of $2^3 - 1$ elements.

Given a completion structure, a node x is blocked if none of its ancestors are blocked, and it has a *witness* x' such that

– $\mathcal{B}(x) = \mathcal{B}(x')$ and $\mathcal{B}'(x) = \mathcal{B}'(x')$

In this case, we say x' blocks x. It is static and is based on propositional-branch equality. For details see below on soundness and completeness.

The *primitive clashes* $\mathcal{ALCQI}$ include any superset of $\{\neg\top\}$, $\{C, \neg C\}$, and $\{\exists^{\leq -1} R.C\}$. The latter is new and is for *fine-tuned modal constraints*. It is reasonable to require that the constants in modal constraints (i.e. qualified number restrictions) are given as non-negative integers. By fine-tuning, possibly it gets a constraint like $\{\exists^{\leq -1} R.C\}$ which we stipulate as trivially unsatisfiable.

[8] For clarity, we purposely do not show GCIs in these rules. However, the rules and the algorithm must take the chunk GCI into consideration.

To generalize primitive clashes, we use the $\perp$-sets originally introduced in [DM99]. Inconsistency inference is performed on demand by tableau procedures.

The following are the *inconsistency propagation rules* for $\perp$-set.

$\perp$-0-rule:		$\{\neg\top\} \in \perp$-sets.
$\perp$-1-rule:		$\{C, \neg C\} \in \perp$-sets.
$\perp$-2-rule:		$\{\exists^{\leq -1} R.C\} \in \perp$-sets.
$\perp$-3-rule:	if	$\alpha \cup \{G, \mathcal{K}_a\} \in \perp$-sets
	then	$\alpha \in \perp$-sets.
$\perp$-4-rule:	if	(1) $\alpha \in \perp$-sets, and
		(2) $\alpha \subseteq \beta$
	then	$\beta \in \perp$-sets.
$\perp$-5-rule:	if	(1) $\alpha \cup \{C\} \in \perp$-sets, and
		(2) $\alpha \cup \{D\} \in \perp$-sets
	then	$\alpha \cup \{C \sqcup D\} \in \perp$-sets.
$\perp$-6-rule:	if	(1) the set of modal constraints about R is $\mathcal{M}$, and
		(2) $\mathcal{M}$'s atom decompositions about R-role-fillers is $\mathcal{D}$, and
		(3) $\mathcal{D}$'s linear-integer-inequalities *lii* is infeasible
	then	$\mathcal{M} \in \perp$-sets.

Fig-2. The inconsistency propagation rule for $\mathcal{ALCQI}$

For jargons in LP/IP, see [Vas83] and [Sch86].

Here is the outline[9] of the intended decision procedure. The decision procedure uses a *restart strategy*[10] and takes a depth-first traversal to construct a tableaux tree. It uses two global data structures. `Nogood` permanently holds triplets like $\langle \mathcal{C}(x), edge2me, \mathcal{B}(x)\rangle$, $\langle \mathcal{C}(x), edge2me, \mathcal{L}(x)\rangle$, and $\langle \emptyset, \epsilon, \mathcal{B}(x)\rangle$ for $\perp$-sets encountered. `Witness` holds intermediate results like $\langle \mathcal{B}(x), \mathcal{B}'(x)\rangle$, and is used for *blocking*. The restart strategy resets `Witness` to empty whenever $\perp$-rules can infer a new `Nogood` element bottom-up. This inconsistency inference is triggered by the primitive clashing or by the (cache) hitting of `Nogood`.

The procedure decides E as *unsatisfiable* if $E \in$ `Nogood`; or otherwise decides E *satisfiable* if the size of `Nogood` is not changed. In other cases, the procedure restarts over and over. The termination is guaranteed since the size of `Nogood` is bounded and each restart will find a new (nontrivial) inconsistency set.

4 Correctness

4.1 Completeness

For the completeness, we need to prove the correctness for what regards concept unsatisfiability. Taking the approach in [DM99], we start with a lemma saying that $\perp$-rules correctly propagate inconsistencies.

Lemma 1. *The $\perp$-rules generate only unsatisfiable sets.*

[9] It will not be presented in this paper due to space limit. For details see [Din07].

[10] The use of restart here is for an easy presentation of the complexity argument.

Proof. By induction on the application of $\perp$-rules.

Base cases. Consider rules $\perp$-0, $\perp$-1, and $\perp$-2. They are clearly unsatisfiable.

Inductive cases. Suppose the claim holds for the antecedent of each $\perp$-rule. We analyze the application of each $\perp$-rule.

- ($\perp$-3): Give C is unsatisfiable w.r.t. G and $\mathcal{K}_a$. Consider that $\top \sqsubseteq G$ and $\top \sqsubseteq \mathcal{K}_a$, in every model for both G and $\mathcal{K}_a$, G and $\mathcal{K}_a$ are equivalent to $\top$. Then it is clear that C is unsatisfiable.
- ($\perp$-4): We prove the claim by contradiction. Suppose $\alpha \subseteq \beta$, α is unsatisfiable and β is satisfiable. Let M be a model for β. Using the sub-model generating technique, there is a sub-model N of M satisfies α, and this contradicts the hypothesis that α is unsatisfiable.
- ($\perp$-5): We prove the claim by contradiction. Suppose $\alpha \sqcap C$ and $\alpha \sqcap D$ are unsatisfiable, but $\alpha \sqcap (C \sqcup D)$ is satisfiable. Let M be a model for $\alpha \sqcap (C \sqcup D)$, then either $\alpha \sqcap C$ or $\alpha \sqcap D$ is satisfied in M. This contradicts the hypothesis.
- ($\perp$-6): The *atom-decomposition* exhaustively generates all combinations of (negated) role fillers on one role R. The *column vector* of the *coefficient matrix* of *lii* takes a value **0** if its corresponding role-filler combination is found *unsatisfiable*; otherwise it remains its initial value. We prove the claim by contradiction. Suppose $\mathcal{M}$ is satisfiable, then this leads to a feasible (conjuncted) combination of role fillers. This contradicts the hypothesis. □

Lemma 2. (Completeness) *If $n \in \perp$-sets, then n is unsatisfiable.*

4.2 Soundness

Denote $\mathbf{T}$ the completed tree constructed. For node $x_i \in \mathbf{T}$, denote its initial label as $\mathcal{L}(x_i)$, its current propositional branch as $\mathcal{B}(x_i)$, and the fine tuned one as $\mathcal{B}'(x_i)$. The algorithm takes a DFS traversal to build $\mathbf{T}$ starting from the root node x_0, and uses the global data structures `Witness` and `Nogood`.

We denote $x_i \lhd x_j$ if x_i is expanded (completed) before x_j does. The blocking relationship conforms to this (node expansion) ordering. Only completed propositional branches enter their pairwise label sets in `Witness`. The blocking nodes must be *propositionally completed* (so that the conventional $\sqcap$-rule and $\sqcup$-rule are no longer applicable.), fine-tuned and not in `Nogood`.

Lemma 3. (Soundness) *If there is tableau tree $\boldsymbol{T}$ for $\mathcal{L}(x_0) = \{E\}$ w.r.t. G and $\mathcal{K}_a$, then there is a model M for $\mathcal{L}(x_0)$ w.r.t. G.*

Proof. It takes three steps.

(1) To admit infinite models, we consider paths in $\mathbf{T}$. The mapping $\mathbf{Tail}(p)$ returns the last element in a path p. Give a path $p = [x_0, ..., x_n]$, where x_i are nodes in $\mathbf{T}$, $\mathbf{Tail}(p)= x_n$. Paths in $\mathbf{T}$ are defined inductively as follows:

- for the root node x_0 in $\mathbf{T}$, $[x_0]$ is a path in $\mathbf{T}$.
- for a path p and a node x_i in $\mathbf{T}$, $[p, x_i]$ is a path in $\mathbf{T}$ iff
  - x_i is not blocked, and

* x_i is a successor of **Tail**(p) and the unknown[11] $v_{x_i} > 0$, or
* y is a successor of **Tail**(p) and x_i blocks y and the unknown $v_y > 0$.

- x_i is not known to be unsat (i.e., its related triplets $\notin$ `Nogood`), and

The pre-model $M' = (\Delta^{\mathcal{I}'}, .^{\mathcal{I}'})$ can be defined with:
$\Delta = \{x_p | \ p \text{ is a path in } \mathbf{T}\ \}$
$x_p \in (\mathcal{L}(\mathbf{Tail}(p)) \sqcap \mathcal{B}(\mathbf{Tail}(p)) \sqcap \mathcal{B}'(\mathbf{Tail}(p)))^{\mathcal{I}}$
$\{\langle x_p, x_q\rangle \mid \langle x_p, x_q\rangle \in (R)^{\mathcal{I}}\} = \{\langle x_p, x_q\rangle \in \Delta \times \Delta | \ q = [p, \mathbf{Tail}\ (q)]$ and
1. **Tail**(q) is an R-successor of **Tail**(p), or
2. $\exists y \in \mathbf{T}$, y is an R-successor of **Tail**(p) and **Tail**(q) blocks y $\}$

$\bigcup$ $\{\langle x_p, x_q\rangle \in \Delta \times \Delta | \ p = [q, \mathbf{Tail}\ (p)]$ and
1. **Tail**(p) is an R^--successor of **Tail**(q), or
2. $\exists y \in \mathbf{T}$, y is an R^--successor of **Tail**(q) and **Tail**(p) blocks y $\}$

(2) Consider the unknown variable v_x that corresponds to each node x of M', duplicate as many $v_x > 0$ numbers of x as the solution requires. This lead to the model $M'' = (\Delta^{\mathcal{I}''}, .^{\mathcal{I}''})$. Each element of M'' is clash-free and is saturated w.r.t. the local cardinality restrictions. M'' is a model for E and G and $\mathcal{K}_a$.

(3) Use the *sub-model generating technique* to extract a model M (for E and G) from M'' (which is for E and $G \cup \mathcal{K}_a$). □

5 Complexity

Lemma 4. (Termination) *The algorithm terminates in $c^{O(n)}$ for some constant $c > 1$, where n is the size of the converted problem.*

Proof. (1) Due to the blocking strategy, the tree size is bounded by $a^{O(n)}$ for some constant $a > 1$. (2) Each node of the tree takes a single exponential cost in n. (3) The size of `Nogood` is bounded by another single exponential function in n. The restart strategy forces at least one new `Nogood` will be inferred when restarting happens. This guarantees at most $\|$`Nogood`$\|$ trees will be constructed. The termination is within $c^{O(n)}$ for some constant $c > 1$. □

Theorem 1. *The tableau-based decision procedure decides $\mathcal{ALCQI}$ concept satisfiability problems in ExpTime in the worst case w.r.t. GCIs.*

6 Summary and Related Work

We have investigated the satisfiability problem in $\mathcal{ALCQI}$ w.r.t. a set of general inclusion axioms and also the applicability problem of the *tableaux caching* technique in *tree structures* restricted by local cardinality constraints and inverse relations. The work is inspired by the ExpTime tableaux procedure given in [DM99]. The topic of tableaux-based reasoning for qualified number restrictions has been well investigated, and it requires a thorough study to distill the contributions as previously made in [OK99] [Tob99] [HTM01] [HS02], and [BHLW03]

[11] Each tableaux node corresponds to one variable of one *lii* at its predecessor node.

[Hla04], and many more on reasoning of finite models. [HST00] shows that the $\mathcal{SHIQ}$ enjoys the *tree model property*, and so does the $\mathcal{ALCQI}$.

We have blurred the distinction between the *blocking technique* and the *tableaux caching*. Regardless of the differences, both are for the termination of tableau procedures. The soundness issue of *tableaux caching* come to the surface with inverse roles for years. There was a tackling of this problem[DH05] with the *precompilation technique*. For an ExpTime procedure on $\mathcal{ALCFI}$, see [Din07].

In summary, we have presented (1) the use of the (restricted) analytic-cut for $\mathcal{ALCQI}$, and (2) a tableau-based method of worst-case ExpTime insensitive to the coding of numbers, and (3) a way to use the tableaux caching technique for a logic having both inverse roles and qualified number restrictions w.r.t. GCIs. For a verbose version giving details of the algorithm see [Din07]. Refinements, empirical issues and optimisations are to be considered in our next work.

Acknowledgements

The author thanks professor Vasek Chvatal for several inspiring discussions.

References

BCM$^+$03. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

BHLW03. F. Baader, J. Hladik, C. Lutz, and F. Wolter. From tableaux to automata for description logics. *LPAR03*, pages 1–32, 2003.

DH05. Yu Ding and Volker Haarlsev. Towards efficient reasoning for description logics with inverse roles. *DL-Workshop'05*, 2005.

Din07. Yu Ding. Tableau-based reasoning for dls with inverse roles revisited. *Technical Report*, pages 1–57, 2007.

DM99. Francesco M. Donini and Fabio Massacci. Exptime tableaux for $\mathcal{ALC}$. *Artificial Intelligence*, 124:87–138, 1999.

Hla04. J. Hladik. A tableau system for the description logic SHIO. *IJCAR04*, 2004.

HS02. Ian Horrocks and Ulrike Sattler. Optimised reasoning for $\mathcal{SHIQ}$. *ECAI 2002*, pages 277–281, 2002.

HST00. Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. *Proc. of the 17th Int. Conf. on Automated Deduction (CADE 2000)*, pages 482–496, 2000.

HTM01. Volker Haarslev, Martina Timmann, and Ralf M"oller. Combining tableaux and algebraic decision procedures for dealing with qualified number restrictions in description logics. *IJCAR-2001*, pages 39–48, 2001.

OK99. H. J. Ohlbach and J. K"ohler. Modal logics, description logics and arithmetic reasoning. *Artificial Intelligence*, 109:1–31, 1999.

Sch86. Alexander Schrijver. *Theory of Linear and Integer Programming*. A Wiley-Interscience Publication, 1986.

Tob99. Stephan Tobies. On the complexity of counting in description logics. *DL Workshop'99*, 1999.

Vas83. Chavatal Vasek. *Linear Programming*. N.Y., W. H. Freeman, 1983.